IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION

EOLAS TECHNOLOGIES, INC.   §
     §
       Plaintiff,   §
     §
vs.   §   **CASE NO. 6:09-CV-446**
     §   **PATENT CASE**
ADOBE SYSTEMS, INC., et al   §
     §
       Defendants.   §

MEMORANDUM OPINION AND ORDER

This Memorandum Opinion construes the disputed terms in U.S. Patent Nos. 5,838,906 (the

"'906 Patent") and 7,599,985 (the "'985 Patent"). The Court further **DENIES** Defendants' Motion

for Partial Summary Judgment of Indefiniteness Under 35 U.S.C. § 112, ¶2 (Docket No. 568).

BACKGROUND

Patents-In-Suit

The '906 Patent, entitled "Distributed Hypermedia Method for Automatically Invoking

External Application Providing Interaction and Display of Embedded Objects Within a Hypermedia

Document," issued on November 17, 1998 to Michael Doyle, David C. Marin, and Cheong S. Ang.

The '985 Patent, entitled "Distributed Hypermedia Method and System for Automatically Invoking

External Application Providing Interaction and Display of Embedded Objects Within a Hypermedia

Document," issued on October 6, 2009 to Michael Doyle, David C. Marin, and Cheong S. Ang. The

'906 and '985 patents are generally directed to a software system that is operable without user

activation to access an object, present it in a browser display window, and then allow a user to

manipulate the object.[1] Eolas is the exclusive licensee of the patents-in-suit.

Eolas alleges that Defendants Adobe Systems, Inc.; Amazon.com, Inc.; Apple, Inc.; CDW Corp.; Citigroup, Inc.; Ebay, Inc.; Frito-Lay, Inc.; The Go Daddy Group, Inc.; Google, Inc.; J.C. Penney Co., Inc.; JPMorgan Chase & Co.; New Frontier Media, Inc.; Office Depot, Inc.; Perot Systems Corp.; Playboy Enterprises Intl, Inc.; Rent-A-Center, Inc.; Staples, Inc.; Oracle America, Inc. f/k/a Sun Microsystems, Inc.; Texas Instruments, Inc.; Yahoo! Inc.; and YouTube, LLC ("Defendants") infringe the '906 and '985 patents.

Prior Litigation Involving The '906 Patent

In December 2000, Eolas filed suit against Microsoft Corporation for infringement of the '906 Patent in the Northern District of Illinois. *Eolas v. Microsoft Corp.*, 2000 U.S. Dist. LEXIS 18886 (N.D. Ill. 2000) ("*Microsoft*"). The *Microsoft* Court construed a term that is disputed in this case, executable application. The *Microsoft* case proceeded to trial and was appealed to the Federal Circuit. On appeal, the Federal Circuit addressed—among other things—construction of the term executable application. *Eolas v. Microsoft Corp.*, 399 F.3d 1325 (Fed. Cir. 2005). Eolas contends this Court should adopt the prior construction of executable application because its construction was upheld by the Federal Circuit.

APPLICABLE LAW

It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude.'" *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (quoting *Innova/Pure Water Inc. v. Safari Water Filtration Sys., Inc.*, 381

---

[1] The patents have largely the same specification. Unless otherwise noted, all citations are to the '906 Patent's specification.

F.3d 1111, 1115 (Fed. Cir. 2004)). In claim construction, courts examine the patent's intrinsic

evidence to define the patented invention's scope. *See id.*; *C.R. Bard, Inc. v. U.S. Surgical Corp.*,

388 F.3d 858, 861 (Fed. Cir. 2004); *Bell Atl. Network Servs., Inc. v. Covad Commc'ns Group, Inc.*,

262 F.3d 1258, 1267 (Fed. Cir. 2001). This intrinsic evidence includes the claims themselves, the

specification, and the prosecution history. *See Phillips*, 415 F.3d at 1314; *C.R. Bard, Inc.*, 388 F.3d

at 861. Courts give claim terms their ordinary and accustomed meaning as understood by one of

ordinary skill in the art at the time of the invention in the context of the entire patent. *Phillips*, 415

F.3d at 1312–13; *Alloc, Inc. v. Int'l Trade Comm'n*, 342 F.3d 1361, 1368 (Fed. Cir. 2003).

The claims themselves provide substantial guidance in determining the meaning of particular

claim terms. *Phillips*, 415 F.3d at 1314. First, a term's context in the asserted claim can be very

instructive. *Id*. Other asserted or unasserted claims can also aid in determining the claim's meaning

because claim terms are typically used consistently throughout the patent. *Id*. Differences among

the claim terms can also assist in understanding a term's meaning. *Id*. For example, when a

dependent claim adds a limitation to an independent claim, it is presumed that the independent claim

does not include the limitation. *Id*. at 1314–15.

"[C]laims 'must be read in view of the specification, of which they are a part.'" *Id*. (quoting

Markman v. Westview Instruments, Inc., 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc)). "[T]he

specification 'is always highly relevant to the claim construction analysis. Usually, it is dispositive;

it is the single best guide to the meaning of a disputed term.'" *Id*. (quoting *Vitronics Corp. v.

Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299

F.3d 1313, 1325 (Fed. Cir. 2002). This is true because a patentee may define his own terms, give

a claim term a different meaning than the term would otherwise possess, or disclaim or disavow the

claim scope. *Phillips*, 415 F.3d at 1316. In these situations, the inventor's lexicography governs. *Id*. Also, the specification may resolve ambiguous claim terms "where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone." *Teleflex, Inc.*, 299 F.3d at 1325. But, "'[a]lthough the specification may aid the court in interpreting the meaning of disputed claim language, particular embodiments and examples appearing in the specification will not generally be read into the claims.'" *Comark Commc'ns, Inc. v. Harris Corp.*, 156 F.3d 1182, 1187 (Fed. Cir. 1998) (quoting *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571 (Fed. Cir. 1988)); *see also Phillips*, 415 F.3d at 1323. The prosecution history is another tool to supply the proper context for claim construction because a patent applicant may also define a term in prosecuting the patent. *Home Diagnostics, Inc., v. Lifescan, Inc.*, 381 F.3d 1352, 1356 (Fed. Cir. 2004) ("As in the case of the specification, a patent applicant may define a term in prosecuting a patent.").

Although extrinsic evidence can be useful, it is "'less significant than the intrinsic record in determining the legally operative meaning of claim language.'" *Phillips*, 415 F.3d at 1317 (quoting *C.R. Bard, Inc.*, 388 F.3d at 862). Technical dictionaries and treatises may help a court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but technical dictionaries and treatises may provide definitions that are too broad or may not be indicative of how the term is used in the patent. *Id*. at 1318. Similarly, expert testimony may aid a court in understanding the underlying technology and determining the particular meaning of a term in the pertinent field, but an expert's conclusory, unsupported assertions as to a term's definition is entirely unhelpful to a court. *Id*. Generally, extrinsic evidence is "less reliable than the patent and its prosecution history in determining how to read claim terms." *Id*.

The patents in suit may contain means-plus-function limitations that require construction. Where a claim limitation is expressed in "means plus function" language and does not recite definite structure in support of its function, the limitation is subject to 35 U.S.C. § 112, ¶ 6. *Braun Med., Inc. v. Abbott Labs.*, 124 F.3d 1419, 1424 (Fed. Cir. 1997). In relevant part, 35 U.S.C. § 112, ¶ 6 mandates that "such a claim limitation 'be construed to cover the corresponding structure . . . described in the specification and equivalents thereof.'" *Id*. (citing 35 U.S.C. § 112, ¶ 6). Accordingly, when faced with means-plus-function limitations, courts "must turn to the written description of the patent to find the structure that corresponds to the means recited in the [limitations]." *Id*.

Construing a means-plus-function limitation involves multiple inquiries. "The first step in construing [a means-plus-function] limitation is a determination of the function of the means-plus-function limitation." *Medtronic, Inc. v. Advanced Cardiovascular Sys., Inc.*, 248 F.3d 1303, 1311 (Fed. Cir. 2001). Once a court has determined the limitation's function, "the next step is to determine the corresponding structure disclosed in the specification and equivalents thereof." *Id*. A "structure disclosed in the specification is 'corresponding' structure only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim." *Id*. Moreover, the focus of the "corresponding structure" inquiry is not merely whether a structure is capable of performing the recited function, but rather whether the corresponding structure is "clearly linked or associated with the [recited] function." *Id*.

<center>**DISPUTED TERMS**</center>

Executable Application

"Executable application" is present in every claim of the patents-in-suit. Eolas contends this term needs no further construction. However, if the Court determines the term requires construction, Eolas proposes "any computer program code, that is not the operating system or a utility, that is launched to enable an end-user to directly interact with the data." Eolas argues that this construction has been affirmed by the Federal Circuit, citing *Eolas Techs., Inc. v. Microsoft Corp.* ("*Microsoft*") 399 F.3d 1325, 1336 (Fed. Cir. 2004). Eolas further contends Defendants' proposed construction improperly imports narrowing limitations.

Defendants contend the term should be construed as "a native binary program that remains separate from the browser and is not part of an operating system or a utility." Thus, the parties disagree about whether the executable application is limited to native binary code or whether it can be other computer program code (such as interpretive code).

Prior Construction in Microsoft

Defendants primarily argue that Plaintiff's *stare decisis* argument is without merit because the court in *Microsoft* did not squarely address this issue in its prior construction, as it was not materially contested.[2] The dispute regarding "executable application" in *Microsoft* centered around whether the application was limited to "standalone programs." *Eolas*, 2000 U.S. Dist. LEXIS at *13–14. In *Microsoft*, Eolas proposed "program code for causing the display of the object and

---

[2] Defendants also argue that: (1) the *Microsoft* construction occurred before the Federal Circuit's *en banc* opinion in *Phillips v. AWH Corp.* (415 F.3d 1303 (Fed. Cir. 2005) (en banc)), and the term was construed too broadly; and (2) the '906 patent has undergone two reexaminations following the *Microsoft* construction. Defendants contend these reexaminations are a significant change to the intrinsic record.

enabling interactive processing of that object." *Id*. Microsoft argued that executable application referred to "standalone" programs only. *Id*. The District Court analyzed the term with regard to whether it included components that were not standalone (e.g. dynamically linked libraries, or "DLLs," such as spell check). *Id*. The Federal Circuit also analyzed this term with regard to standalone programs on appeal. *Eolas Techs*., 399 F.3d at 1336.

Here, the parties' dispute does not involve standalone programs. Instead, the parties disagree about whether the program is limited to native binary—a compiled program that is in machine code—as opposed to an interpretable program that is in, for example, bytecode (e.g. scripts). Accordingly, the dispute in this case was not raised in *Microsoft* and Plaintiff's *stare decisis* argument is misplaced. The Court, therefore, declines to adopt Plaintiff's proposed construction based on this argument.

Patents-in-suit and Prosecution History

The term "executable application" is not used in the written description of the '906 patent specification, it only appears in the claims. The specification uses the terms "controllable application" in the summary at 6:58–59, "application program" at 7:3, and "application client" in numerous places including describing the embodiment of Fig. 5 at 8:67.

A review of the prosecution history explains the difference in terminology between the specification and the claims, and also gives insight into the applicant's intentions regarding the type of code claimed—interpretive or compiled. In the claims as originally filed, for example claim 1, the applicant used the term "controllable application." *See* Defendants' Exh. J at 1 and appended Mukherjee Declaration at 906 PH Ex. 1, PH_001_783829. First, in rejecting claims 1–4 and 15–16

as being anticipated under 35 U.S.C. § 102, the examiner characterized USC's Mercury Project prior

art as using a method as claimed wherein a hypermedia document included an embedded controllable

application. Second, in rejecting claims 1–43 as being obvious under 35 U.S.C. § 103, the examiner

relied upon a combination of prior art disclosed by the applicant and Hansen.[1] In fashioning the

obviousness rejection, the examiner considered the applicant's disclosed prior art to be without an

embedded "controllable application" in the hypermedia document. The scope of the construction

given by the examiner to the term "controllable application" was inclusive of

"executable/interpretable/launchable program instructions/codes." Hansen was alleged to teach

enhancing hypertext documents by embedding programs into such documents.

In response to the rejections, the applicant amended claim 1, for example, to redirect focus

from displaying a hypermedia document having an embedded controllable application to focusing

on the browser display of a hypermedia document having an embed text format that leads to

identification and locating of an "executable application." The applicant also added computer

program product claims reciting "an [external] executable application" in, for example, claim 44.

In dependent claim 45, the applicant reintroduced "controllable application" as being a further

characterization of "executable application."[2]

In remarks to the examiner, the applicant pointed out that, as amended, claim 1 recited an

executable application invoked by the browser to display and process an object within the browser's

---

[1] A listing of prior art cited by applicant in its Information Disclosure Statement ("IDS") is shown at Defendants' Exh. J at 1 and appended Mukherjee Declaration at 906 PH Ex. 2, PH_001_783875-77.
[2] In claims that dependent from claim 45 (i.e., claims 50 and 52), the applicant further specified that the controllable "executable" application is a spreadsheet program or a word processor. Spreadsheet and word processor programs are compiled programs in machine code.

window. The applicant distinguished the Mercury Project prior art because it involved a new server-generated webpage that is sent to the browser for display in place of an original webpage being displayed in the browser window, i.e., static HTML document display. In contrast, the executable application processes an object and displays it in a portion of the browser window. Hansen was distinguished as having both an *executable script* (i.e., program code)[3] and an object to be embedded within a document rather than a browser-invoked external application.

Thus, beginning with the examiner's construction of "controllable application" as being inclusive of the separate types of program instructions or code known as "executable" and "interpretable," the applicant redirected the claims to specify an "executable application." Further, the applicant specified only particular types of "executable applications" in further restricting the "executable application" limitation to a spreadsheet program or a word processor program. In characterizing the prior art, the applicant observed a known distinction between an "executable application" and an "executable script."

The Microsoft Press Dictionary defines the term "executable program" as follows:

A computer program that is ready to run. The term usually refers to a compiled program that has been translated into machine code in a format that can be loaded into memory and run. However, for interpreted languages it can simply refer to source code in the proper format. Applications such as word-processing programs are executable programs. The user does not have to alter the program in any way before being able to run it.

Microsoft Press Dictionary (1991). Thus, the term "executable program (application)" is primarily used to designate a compiled program in machine code that can be loaded into memory and run, but

---

[3] "Scripts" are known to be "interpretable" programs.

can also reference source code when used in the context of interpretable programs.

While the applicant's amendments to claim 1 in response to the examiner's rejection, and the applicant's remarks as to the Mercury Project prior art standing alone may not evince a clear disavowal of interpretable programs from the meaning of "executable application (program)," the primary and secondary meanings of the term indicated by the Microsoft Press Computer Dictionary definition set out above are substantiated and followed by the applicant in this portion of the prosecution history.

Finally, turning to the specification of the '906 patent, Fig. 5 illustrates an embodiment that includes application client 210 described as residing on the hard disk or RAM of the client computer. The application client 210 is further described as being loaded and executed when the browser detects a link to it. *See* 8:66–9:7. The description refers to the operation of a computer processing unit (CPU) to execute or run program instructions in machine code that have been moved from non-volatile memory to random access memory by an operating system. The CPU fetches a machine code instruction from memory according to a specified address. The machine code instruction must be in a binary form that is compatible with the native instruction set of the CPU for operations (i.e., OP code). Further, examples of application client 210 are identified as spreadsheet program, database program, and word processor. *See* 13:11–15. As set forth in the Microsoft Computer Dictionary definition, such programs are "executable" meaning that they are compiled into machine code and ready to be run directly on the computer processor (CPU).[4]

---

[4] *See* Defendants' Ex. J at 18-19; Notice of Intent to Issue Ex Parte Reexamination Certificate. The examiner's comments are relied upon for their technical description of the computer technology related to the '906 patent. PTO examiners are assumed to have some expertise in and be familiar from their work with the technology of the art they examine. *Aloft Media, LLC v. Adobe Systems, Inc.,* 570 F. Supp. 2d 887 (E.D. TX. 2008).

The '906 patent claims also recite that the client workstation is "executing" or is caused "to execute" a browser application. *See* '906 Patent Reexamination Certificate, claim 1 at 1:31–32 and claim 6 at 3:56–57. The '906 patent describes that the browser client 208 of the preferred embodiment is Mosaic. *See* '906 patent at 8:9–12, 10:16, 14:9–11, and 16:12. As the applicant explained to the PTO, Mosaic browser subroutines that responded to the meanings of tags "were built from machine-specific native code . . . ." *See* Mukherjee Declaration at 906 PH Ex. 16, PH_001_784146–47. Thus, the "executing" and "to execute" claim language as to the browser application, when read in view of the intrinsic record, demonstrates that the applications (browser and executable) are compiled programs in machine code.[5]

Accordingly, the Court construes the term "executable application" in accordance with the primary sense of the Microsoft Press Dictionary definition as "a compiled program that is in native machine code in a format that can be loaded into memory and run."

Automatically Invoking the Executable Application

"Automatically [invoking/invoke] [the/said] executable application" and "executable application is automatically invoked by the browser" are collectively present in every claim of the patents-in-suit. Eolas contends these terms need no construction. Alternatively, Eolas offers the following: "automatically calling or activating the executable application" and "executable application is automatically called or activated by the browser." Defendants' revised construction proposes "the executable application is launched to permit a user to interact with the object without

---

[5] In addition, the applicant made the point that by having the executable application execute on the client workstation, the browser's functionality can be extended *without changes to the browser's object code that is being executed*. *See*, Mukherjee Declaration at 906 PH Ex. 11, PH_001_784052.

any intervening activation of the object by the user." The dispute centers around whether automatically invoking the executable application excludes an application that requires a "click" or activation by the user in order to launch. Plaintiff argues that Defendants' "without any intervening activation of the object by the user" language constitutes an improper limitation that is not supported by the intrinsic record.

Defendants rely on portions of the prosecution history, arguing that the term was added to the claims in order to distinguish the prior art Mosaic browser. Specifically, Defendants contend that during prosecution, the applicant distinguished the Mosaic browser by noting that Mosaic required a user to first "click" on a hyperlink URL after a hypermedia document has been parsed in order for the user to be able to interact with an object. Thus, Defendants argue that the term requires this construction to prevent Plaintiff from recapturing what it surrendered to secure allowance of the claims.

The prosecution history relied on by Defendants concerns the applicant's response to an obviousness rejection based upon a combination of references including the prior art Mosaic browser. *See* Defendants' Exh. J at 1 and appended Mukherjee Declaration at 906 PH Ex. 10, PH_001_784013-16. Specifically, the applicant amended the claims to recite "automatically invoke said executable application." *See* Defendants' Exh. J at 1 and appended Mukherjee Declaration at 906 PH Ex. 11, PH_001_784030. In remarks to the examiner, the applicant characterized the Mosaic browser as launching viewer applications in a separate window in response to a user's interactive command (i.e., "clicking" on a URL link). *Id.* at 784034–35. Thus, the applicant distinguished the Mosaic browser: "[a]ccordingly, [in Mosaic] the external application is not automatically invoked as a result of the browser parsing the hypermedia document text, as required by the claim, but rather

it is invoked by an interactive command given by the user, namely interactively selecting the URL anchor." *Id.* at 784039.

As Plaintiff argues, the prosecution history does not support a construction of the term "automatically invoke" that imports a limitation of "activation of the object." However, the prosecution history does support a construction that an executable application is launched without user activation.[6] Accordingly, the Court construes the terms "Automatically [invoking/invoke] [the/said] executable application" and "executable application is automatically invoked by the browser" as meaning that "the executable application is launched without user activation."

Text Format

"Text format" is present in every claim of the patents-in-suit. Eolas contends the term means "text that initiates processing," while Defendants contend it means "tags or symbols that specify document formatting." The parties dispute whether the term should be limited to tags or symbols, or whether it can include text. Defendants argue that a "text format" is a tag or symbol and not simply text. Defendants contend that Eolas's construction reads "formats" out of the claim. Eolas points to the specification—which includes tags, symbols, and words—to demonstrate that Defendants' proposed construction improperly imports limitations into the claim. Tr. 46, Ll. 21–23.

Plaintiff's proposed construction improperly sets forth what a "text format" allegedly does and not what it is. Defendants' proposed construction is too limiting. At the claim construction hearing, the Court proposed "coded information that describes how the content of a hypermedia document is to be interpreted by a browser application for display." Tr. 48:19–25. The parties

---

[6] Indeed, the claim language itself sets forth that the browser launches the executable application when parsing of the hypermedia document reveals an embed text format. Such was the argument made during prosecution to distinguish Mosaic.

requested 48 hours to review this proposed construction prior to committing to it. At the hearing, Defendants requested "interpreted" be changed to "used," citing the specification's teaching "that the browser directly uses text formats, tags or symbols to control how that document looks on the display." Tr. 51: 2–5.

Following the hearing, both parties submitted additional proposed constructions along with arguments in support. Defendants' new proposed instruction is "coded information that describes how the content of a hypermedia document is to be displayed by a browser application." Defendants' proposed construction deletes the word "interpreted," as requested at the hearing. Plaintiff agrees with the Court's proposed construction except for the language "for display." Plaintiff argues this language is improperly limiting. Plaintiff's proposed construction is "coded information that describes how the content of a hypermedia document is to be interpreted by a browser application." Upon review of the parties' additional arguments and proposed constructions, the Court construes the term as "coded information that describes how the content of a hypermedia document is to be displayed by a browser application."

Embed Text Format/Embed Text Format Specifies the Location of at least a portion of [an/said] object/Embed Text Format Located at a first location in said first distributed hypermedia document/Embed Text Format [which] correspond[s/ing] to [a/said] first location in the document/identify[ing] and embed text format

These "Embed Text Format" terms are collectively present in every claim of the patents-in-suit. Eolas contends none of these terms need construction. However, Eolas alternatively proposes constructions for each term should the Court determine construction is necessary.

Embed Text Format

Eolas contends "embed" has no technical meaning and no specialized meaning within the

patent and thus should be given its plain and ordinary meaning. Alternatively, if the Court

determines that a construction is needed, Eolas offers "text format for embedding an object."

Defendants propose "a tag that specifies the object to be embedded at the location of the tag."

Defendants contend "embed text format" is not a term of art, nor is it found in the written description

or original claims. The term was coined by the inventor and added by amendment in August 1996

to designate a special tag called the EMBED tag. Def. Br. At 11. Plaintiff argues Defendants'

construction is too limiting for several reasons. First, Plaintiff points to the specification to

demonstrate that a text format can be at least a "word, tag, or symbol." '906 patent at 14:24–29.

Second, Plaintiff argues that pursuant to the doctrine of claim differentiation, the claim must mean

more than a tag. Claim 1 of the '985 patent contains the term at issue. Dependent claim 3 adds the

limitation "where the text formats are HTML tags." '985 patent, claims 1, 3. This gives rise to a

strong presumption that the independent claim is not limited to a tag.

Embed text format designates a special type of text format. The term is supported in the

specification by the disclosure of the EMBED tag ('906 patent, 14:32–33); however, the term is not

limited to only a tag. Consistent with the Court's construction of "text format," and in view of the

specification, the Court construes "embed text format" as "coded information that specifies to a

browser application that an object is to be embedded in a displayed hypermedia document."

Embed Text Format Specifies the Location of at least a portion of [an/said] object

Eolas argues this phrase needs no construction; however alternatively offers "embed text

format that specifies the location of at least part of an object." Defendants propose "specifies the

location of at least a portion of [an/said] object" where "specifies" has its common meaning: "to

name or state explicitly or in detail." The parties appear to agree that the location of at least a

15

portion of an object is specified.

In light of the Court's construction of "text format," this phrase needs no construction. The additional language in this phrase is clear in the context of the claims and will be readily understandable to the jury. *See Orion IP, LLC v. Staples, Inc.*, 406 F. Supp. 2d 717, 738 (E.D. Tex. 2005) (Davis, J.) (stating that "although every word used in a claim has meaning, not every word requires construction" in declining to construe claim terms).; *see also Eon*, 2010 U.S. Dist. LEXIS 83442 at *71–72; *Mirror Worlds*, 2010 U.S. Dist. LEXIS 82070, at *20. Regarding the "specifies the location" language, the parties' proposed constructions are merely attempts to restate the claim language or an effort to include the plain and ordinary meaning. Accordingly, this term needs no construction.

Embed Text Format, Located At/Corresponding To A First Location In Said First Distributed Hypermedia Document

Eolas argues these phrases need no construction; however alternatively offers "embed text format locates at a first location in the first distributed hypermedia document" for "located at a first location." Eolas offers "embed text format which relates to a first location in the document" for "corresponding to a first location." For the "located at a first location" phrase, Defendants offer "tag located at the place in the received document where the embedded object will appear within the displayed document." For the "corresponding to a first location" phrase, Defendants offer "tag located at the place in the received file where the embedded object will appear within the displayed document."

The parties dispute whether the recited phrases "located at a first location" and "corresponding to a first location" mean that the embed text format is located in the document where

the embedded object will appear. Eolas argues that nothing in the ordinary meaning of these phrases requires that location to be "where the embedded object will appear." Defendants respond that the claim language itself specifies that the embed text format and the display area are both at the same location.

This phrase does not require additional construction beyond the Court's construction of "embed text format." The "location" terms are clear and understandable. Defendants' proposed construction improperly includes a requirement that the term designate where the embedded object will appear within the displayed document. This proposed construction accrues based on a limitation that is later in the claim language. The claim goes on to recite ". . . in order to display said object . . . within a display area created at *said first location* . . . ." ('906 Patent, Claim 1 at 17:24–26). This later limitation logically demonstrates that the embed text format location in the document is where the displayed object will appear. However, that requirement does not accrue from the language "at a first location" or "corresponds to a first location." These phrases need no further construction.

Identifying An Embed Text Format/An Embed Text Format Is Identified

Eolas contends no construction is needed; however, it alternatively proposes "detecting an embed text format" and "an embed text format is detected" respectively. Defendants offer "detecting an embed text format during parsing of a hypermedia document" for both phrases. Eolas argues Defendants' proposal improperly equates "identifying" with "parsing," and the doctrine of claim differentiation precludes such an interpretation. Eolas points to claim 1 of the '985 patent—which contains the "identifying an embed text format" term—then notes that dependent claim 5 adds the limitation "where the step of identifying an embed text format comprises: parsing the received file to identify text formats included in the received file." '985 Patent, Claims 1, 5. Thus, Eolas argues

17

that while identification *may* occur during parsing, it is not required.

Defendants argue that the specification describes identifying an embed text format as a sub-step of parsing, and so identifying the embed text format must occur during parsing. Defendants representation of the specification is inaccurate. The specification describes a routine where HTML tags can be identified. The routine (*see* Figure 7A below) has a loop with steps 254, 256, and 258 where step 256 is a parsing step and 258 is an EMBED tag detection step. The step of detecting follows the step of parsing. Thus, the specification does not support Defendants' proposed construction.

In addition, dependent claim 5 narrows the step of identifying to include a step of parsing the file to breakdown the file into the text formats present in the file. Defendants' proposed construction is overly narrow. The term "identifying" is really used in the sense of "determining the existence of." The language is clear when given its ordinary meaning, and no construction is necessary.
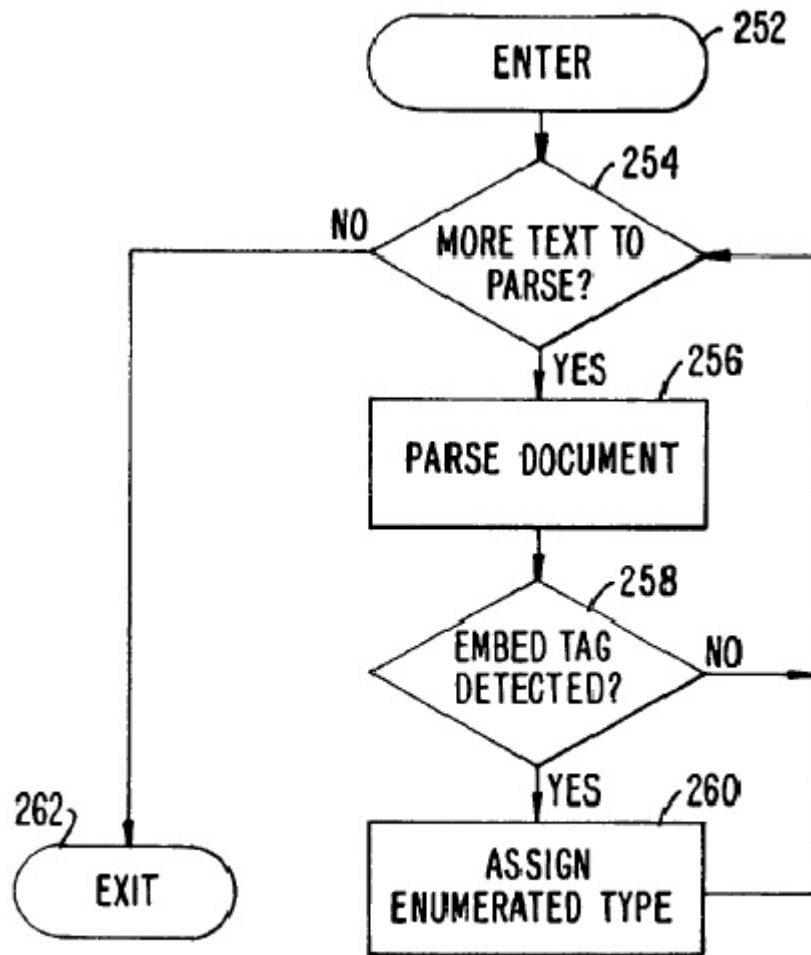
FIG. 7A.

Object

This term is present in every claim of the patents-in-suit. The parties agree that an "object" encompasses "information presentable to a user of a computer system." The dispute centers around whether the object may include a software program or anything with source code or byte code. Eolas proposes "text, images, sound files, video data, documents or other types of information that is presentable to a user of a computer system." Defendants propose "information presentable to a user of a computer system, which is not a program and which does not include source code or byte code."

19

Defendants argue that a software program cannot be an object because it is not information that is presentable to a user for viewing and manipulation. Def. Br. at 18.

Eolas's proposed construction is supported by—indeed comes straight from—the specification. '985 Patent, 2:12–15 ("Objects may be text, images, sound files, video data, documents or other types of information that is presentable to a user of a computer system."). Defendants' proposed construction improperly includes a negative limitation. Eolas's construction could be more clear, and the Court construes the term as, "text, images, sound files, video data, documents, and/or other types of information that is presentable to a user of a computer system."

Hypermedia Document/Distributed Hypermedia Document/File Containing information

These terms are collectively present in every claim of the patents-in-suit. Eolas contends "[first] hypermedia document" should be construed as "a document that allows a user to click on images, sound icons, video icons, etc., that link to other objects of various media types, such as additional graphics, sound video, text, or hypermedia or hypertext documents." For "distributed hypermedia document," Eolas proposes "[first] hypermedia document that allows a user to access a remote data object over a network." Finally, with regard to "file containing information," Eolas proposes "the file contains information to allow the browser application to display at least part of a distributed hypermedia document."

Defendants propose one construction for both "[first] hypermedia document" and "distributed hypermedia document"— "a document received by the browser that includes hyperlinks to graphics, sound, video or other media." For "file containing information," Defendants propose "the file contains information to allow the browser application to display at least part of a distributed hypermedia document."

"[First] hypermedia document" and "distributed hypermedia document" are used interchangeably in the claims and will therefore receive the same construction. Eolas's proposal comes directly from the specification. *See, e.g.* '906 Patent, 2:24–27. Defendants' proposal includes inherent aspects of other surrounding claim limitations. Defendants' requirement that the document be "received by the browser" that includes "links" imports limitations that are unsupported by the intrinsic record. Accordingly, the Court construes "[first] hypermedia document" and "[first] distributed hypermedia document" as "a document that allows a user to click on images, sound icons, video icons, etc. that link to other objects of various media types, such as additional graphics, sound video, text or hypermedia or hypertext documents."

The "file containing information" term does not need further construction. When read in view of the Court's construction of "[first] hypermedia document," the term will be understood by a jury. *See Eon*, 2010 U.S. Dist. LEXIS 83442 at *71–72; *Mirror Worlds*, 2010 U.S. Dist. LEXIS 82070, at *20. The parties' proposed constructions do not provide any additional meaningful guidance beyond the claim language.

Distributed Application

This claim is present in independent claims 36, 40, and 44 of the '985 patent. Both parties agree that a distributed application is an application that is broken up and performed between two or more computers. The parties disagree about: (1) whether an application must be broken up and performed between two or more computers; and (2) whether the actions must be performed in parallel. Eolas proposes "an application that may be broken up and performed among two or more computers." Defendants propose "an application in which tasks are broken up and performed in parallel on two or more computers."

With regard to the first dispute, at the hearing the Court asked the parties whether they would agree to changing Eolas's proposed "may" to "is capable of." Eolas agreed with the Court's proposal. Defendants argued the term should be "that is" not "may." The Court gave the parties an opportunity post-hearing to review the Court's proposed construction and notify the Court as to whether they were amenable to the proposed language. The parties did not change their positions regarding the term. Eolas's proposed definition again comes directly from the specification. '985 Patent, 11:2–4 ("In the present example, tasks such as volume rendering *may be* broken up and easily performed among two or more computers.") (emphasis added). Defendants' requirement that the tasks be broken up is a limitation that is not inherent in the term.

Defendants next argue that the tasks must be performed in parallel (i.e. simultaneously). For support, Defendants cite '985 at 5:31–6:15. This section of the specification describes a specific type of image processing wherein several computers are used for distributed processing. *Id*. However, that description does not specifically relate to the term "distributed application." Defendants' proposed construction improperly brings into the construction one aspect of how the distributed application is performed among two or more computers, i.e. in parallel.

The Court construes the term as "an application that is capable of being broken up and performed among two or more computers."

Client Workstation/Network Server

These terms are collectively present in every claim of the patents-in-suit. Eolas asserts that "client workstation" means "a computer system connected to a network that serves the role of an information requester. Eolas contends "network server" should be construed as "a computer system that serves the role of an information provider." Eolas points to the specification in support of these

constructions. '906 Patent, 4:55–59 ("For purposes of this specification, client and server computers are categorized in terms of their predominant role as either an information requestor or provider. Clients are generally information requesters, while servers are generally information providers.").

Defendants contend "client workstation" means "a desktop or deskside computer with an operating system and hardware designed for technical or scientific applications that provides higher performance than a personal computer." For "network server," Defendants propose "a computer running software that is capable of executing applications responsive to requests from a client workstation, and that processes commands from a client workstation to locate and retrieve documents or files from storage." Defendants argue that workstation is a term of art and implies hardware with greater capacity than a personal computer. Defendants contend the network server was identified as being server computer 204, and the specification describes server computer 204 as having the capabilities set forth in Defendants' proposed construction.

Plaintiff's constructions of "client workstation" and "network server" are consistent with the specification. With regard to Defendants' proposed construction of "client workstation," Defendants' distinction between a workstation and a personal computer is too limiting. Further, the characterization of "higher performance than a personal computer" is subjective and thus indefinite. With regard to "network server," Defendants' proposed construction restricts the term to the preferred embodiment. Accordingly, the Court construes "client workstation" as a computer system connected to a network that serves the role of an information requester," and "network server" as "a computer system that serves the role of an information provider."

Means-Plus-Function Terms

Defendants argue that there are fifty-three claim elements that invoke Section 112, Paragraph 6. These include four groups of claims: (1) '985 claims 24–27: "the method comprising: enabling . . ."; (2) '985 claims 20–23 and 40–43: "communicating . . . in order to cause"; (3) '985 claims 16–19 and 28–31: "software . . . operable to . . . cause"; and (4) '906 claims 6–8 and 13: "computer readable program code for causing . . . ."

Defendants admit none of these claims use the word "means." This fact triggers a presumption that § 112, ¶ 6 does not apply. *Lighting World, Inc. v. Birchwood Lighting, Inc.*, 382 F.3d 1354, 1358 (Fed. Cir. 2004). However, Defendants argue that these terms claim a functional result rather than sufficient structures or acts for achieving that result and thus § 112, ¶ 6 should apply. Assuming § 112, ¶ 6 applies, Defendants argue Eolas failed to identify the proper corresponding structure(s).

Eolas argues that virtually every element asserted by Defendants includes a phrase containing either the words "computer readable program code for . . ." or "software comprising computer executable instructions [to] . . ." followed by a description of the code's (or software's) operation. Pl. Br. at 26. The code and software in these claims describes sufficient structure to avoid the application of § 112, ¶ 6. Defendants present no valid basis for applying § 112, ¶ 6, and the Court declines to do so.

MOTION FOR PARTIAL SUMMARY JUDGMENT OF INDEFINITENESS

Defendants contend claim 28 of the '985 patent and claims 6–8 and 13 of the '906 patent are invalid as indefinite (Docket No. 568). Having considered the motion and all responses, the motion is **DENIED**.

BACKGROUND AND THE PARTIES' CONTENTIONS

Defendants argue these claims are indefinite under the Federal Circuit's holding in *IPXL Holdings, LLC v. Amazon.com, Inc.* because they are drafted to cover both an apparatus and various method steps of using the apparatus. 430 F.3d 1377, 1384 (Fed. Cir. 2005). Specifically, independent claim 6 of the '906 Patent—and dependent claims 7–8 and 13—are directed to both an apparatus ("a computer program product") and to various method steps (e.g. ". . . wherein said object has type information associated with it utilized by said browser to identify and locate . . ."; ". . . wherein said embed text format is parsed by said browser to automatically invoke said executable application . . ."). '906 Patent C2, Claim 6, 4:8–15. Similarly, claim 28 of the '985 patent is directed to both an apparatus ("an executable application") and various method steps performed by a separate "browser application" (e.g. ". . . wherein the executable application is automatically invoked by the browser . . ."). '985 Patent Claim 28, 20:30–32.

Defendants also argue that if §112, ¶ 6 does not apply—and the Court holds that it does not—then the claims are indefinite because they use purely functional language. Specifically, the language in the patents-in-suit "in order to cause," "software . . . operable to . . . cause," and "for causing" is functional.

Plaintiff responds the claims at issue are not mixed method/apparatus claims under *IPXL*. Rather they are "computer readable" code claims. In response to Defendants' functional language

argument, Plaintiff argues that the claims describe capability of the claimed apparatuses, they do not claim the activity itself.

APPLICABLE LAW

A party seeking to invalidate a patent must overcome a presumption that the patent is valid. *See* 35 U.S.C. § 282; *United States Gypsum Co. v. National Gypsum Co.,* 74 F.3d 1209, 1212 (Fed. Cir. 1996); *Hibritech Inc. v. Monoclonal Antibodies, Inc.,* 802 F.2d 1367, 1375 (Fed. Cir. 1986). This presumption places the burden on the challenging party to prove the patent's invalidity by clear and convincing evidence. *Microsoft Corp. v. i4i Limited Partnership*, 131 S. Ct. 2238, 2242 (2011); United *States Gypsum Co.,* 74 F.3d at 1212. Close questions of indefiniteness "are properly resolved in favor of the patentee." *Datamize, LLC v. Plumtree Software, Inc.*, 417 F.3d 1342, 1348 (Fed. Cir. 2005); *Exxon Research & Eng'g Co. v. United States*, 265 F.3d 1371, 1380 (Fed. Cir. 2001).

35 U.S.C. § 112 requires that claims be particular and distinct. "The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention." 35 U.S.C. § 112 ¶ 2. The primary purpose of the definiteness requirement is to ensure public notice of the scope of the patentee's legal protection, such that interested members of the public can determine whether or not they infringe. *Halliburton Energy Servs., Inc. v. M-I, LLC*, 514 F.3d 1244, 1249 (Fed. Cir. 2008). Thus, the definiteness inquiry focuses on how a skilled artisan understands the claims, and a claim is indefinite if the "accused infringer shows by clear and convincing evidence that a skilled artisan could not discern the boundaries of the claim based on the claim language, the specification, and the prosecution history, as well as her knowledge of the relevant art area." *Id.* at 1249–50. "If the meaning of the claim is discernible, even though the task may be formidable and the conclusion may be one over

which reasonable persons will disagree, . . . the claim [is] sufficiently clear to avoid invalidity on indefiniteness grounds." *Exxon Res. & Eng'g Co. v. United States*, 265 F.3d 1371, 1375 (Fed. Cir. 2001). Thus, a claim is indefinite only if its meaning and scope are "insolubly ambiguous." *Id.*

A claim may be invalid if it combines two separate statutory classes of invention. *See IPXL Holdings, L.L. C. v. Amazon. com, Inc.,* 430 F.3d 1377, 1384 (Fed. Cir. 2005) (hereinafter "IPXL Holdings"). In *Ex parte Lyell,* the Board of Patent Appeals and Interferences held that reciting both an apparatus and a method of using that apparatus renders a claim indefinite under § 112 ¶ 2. 17 U.S.P.Q.2d 1548, 1550 (B.P.A.I. 1990). On the basis of this rule, the BPAI invalidated as indefinite a claim whose preamble described "an automatic transmission tool . . . and method for using same."

In *IPXL Holdings v. Amazon.com,* the Federal Circuit endorsed the reasoning of *Ex parte Lyell* by invalidating the following claim:

> The system of claim 2 [including an input means] wherein the predicted transaction information comprises both a transaction type and transaction parameters associated with that transaction type, *and the user uses* the input means to either change the predicted transaction information or accept the displayed transaction type and transaction parameters.

Id. (quoting patent) (emphasis added). The Federal Circuit explained that, because the claim recited use by a user as a claim limitation, it is unclear if infringement would occur when one creates a system that allows a user to change the predicted transaction information, or when a user actually uses the system. In other words, the claim language ambiguously claimed both a system and the method for using the system. *Id.* A potential seller of the claimed system would not know from the claim whether it might also be liable for contributory infringement if a buyer later performs the claimed method of using the system. Thus, the Federal Circuit found the claim invalid under § 112

¶ 2. *Id.*

The Federal Circuit recently clarified its holding in *IPXL Holdings*. In *Microprocessor Enhancement Corp. v. Texas Instruments Inc.*, the Federal Circuit explained that apparatus claims are not necessarily indefinite for using functional language. 520 F.3d 1367, 1375 (Fed. Cir. 2008) (citing *Halliburton Energy Servs. v. M-I LLC*, 514 F.3d 1244, 1255 (Fed. Cir. 2008)). For example, functional language which merely describes the capability of the claimed invention will not render a claim invalid under *IPXL Holdings*. *Id.* This sort of language does not impermissibly mix two different statutory classes of subject matter. *See Yodlee, Inc. v. Cashedge, Inc.*, No. 05–01550, 2006 WL 3456610, at *4–6, (N.D. Cal. Nov. 29, 2006) (hereinafter "Yodlee") (finding that the claims at issue are valid because they "describe what the apparatuses do, when used in a certain way. They do not claim *use* of the apparatuses"). Since evaluating indefiniteness arguments under *IPXL Holdings*, courts have repeatedly distinguished *IPXL Holdings* from situations in which the claim language offered sufficient notice to potential defendants as to the actions which would constitute infringement. *See, e.g., Yodlee*, 2006 WL 3456610, at *4; *Toshiba Corp. v. Juniper Networks, Inc.*, No. 03-1035, 2006 WL 1788479, at *4 (D. Del. June 28, 2006); *Duhn Oil Tool, Inc. v. Cooper Cameron Corp.*, 474 F. Supp. 2d 1148, 1164 (E.D. Cal. 2007).

ANALYSIS

A.   The Mixed Method/Apparatus Argument

Defendants argue that the claims at issue are mixed method/apparatus claims and are therefore indefinite under *IPXL*. With regard to claim 28 of the '985 patent, Defendants contend that because the claim covers both an apparatus ("executable application") and method steps that must

be performed by a separate browser, it is unclear whether infringement would occur upon making/using/selling the accused executable application, or only after the separate "browser" has used the executable application to perform the claimed method steps. Similarly, because claim 6 of the '906 patent (as well as the dependent claims at issue) covers an apparatus ("a computer program product"), but all of the method steps in the claim must be performed by a "browser," it is unclear whether infringement occurs upon making/using/selling the accused website, or only after the separate "browser" has used the website to perform the claimed method steps.

Reading the claims in light of cases applying *IPXL Holdings*, however, suggests that Defendants' position is misplaced. *Yodlee*, 2006 WL 3456610, at *5 (citing '073 patent at 10:2–13 and 11:11–24); *see also Toshiba Corp. v. Juniper Networks, Inc.*, 2006 WL 1788479, at *4 (evaluating the claim term "wherein the control message processing unit communicates with the next hop node . . . ."). It is well-established that for a limitation to introduce a method step, the limitation must require action, or "actual use" of something instead of merely requiring or setting forth a particular capability. *See, e.g., Microprocessor Enhancement*, 520 F.3d at 1374–75; *compare with IPXL Holdings*, 430 F.3d at 1384 (invalidating the challenged claim because the user of the apparatus performed the claimed method of using the apparatus). The claim limitations challenged in this case do not require action. Instead, they delineate the capability of the claimed apparatuses.

Claim 28 of the '985 patent reads:

> One or more **computer readable media encoded with software comprising an executable application** for use in a system having at least one client workstation and one network server coupled to a network environment, **operable to**:
>
> cause the client workstation to display an object and enable an end-user to directly interact with said object while the object is being displayed within a display area

created at a first location within a portion of a hypermedia document being displayed in a browser-controlled window . . . wherein said client workstation executes said browser application . . . wherein the embed text format which corresponds to said first location in the document is identified by the browser... wherein the type information is utilized by the browsers to identify and locate said executable application, and wherein the executable application is automatically invoked by the browser . . . ."

'985 patent at 20:1–32 (claim 28) (emphasis added). The "operable to" language in the claim represents functional limitations of the capabilities of the apparatus, not method steps. *See Finjan, Inc. Secure Computing Corp*., 626 F.3d 1197 (Fed. Cir. 2010) (holding similar claimed language, "computer-readable storage medium storing program code for causing a server . . ." claims capability); *see also Fantasy Sports Props. v. Sportsline.com*, 287 F.3d 1108, 1117–1118 (Fed. Cir. 2002) ("in order to infringe the '603 patent, the code underlying an accused fantasy football game must be written in such a way as to enable a user of that software to utilize the function . . . without having to modify the code.").

Claim 6 of the '906 patent reads:

A **computer program product** for use in a system having at least one client workstation and one network server coupled to said network environment . . . the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein, said computer product further comprising: **computer readable program code for causing** said client workstation to execute a browser application to parse a first distributed hypermedia document to identify text formats included in said distributed hypermedia document and to respond to predetermined text formats to initiate processes specified by said text formats;

computer readable program code for causing said client workstation to utilize said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein the portion of said first hypermedia document is displayed within a first browser-controlled window on said client workstation. . . wherein said object has type information associated with it utilized by said browser to identify and locate an executable application external

30

to he first distributed hypermedia document, and wherein said embed text format is parsed by said browser to automatically invoke said executable application to execute on said client workstation in order to display said object and enable an end-user to directly interact with said object . . . ."

'906 Patent C2 at 3:48–4:20 (claim 6) (emphasis added). The wherein clauses in these claims are clearly linked with the preceding "for causing" language—they further describe the capabilities of the claimed computer program code in its relevant environment.

Other courts have considered indefiniteness arguments similar to those presented herein, and have found that functional apparatus language is not indefinite when it describes the capabilities of the apparatus. *See generally Microprocessor Enhancement*, 520 F.3d at 1374–75; *see, e.g.,Yodlee*, 2006 WL 3456610, at *4; *Toshiba Corp. v. Juniper Networks, Inc.*, No. 03-1035, 2006 WL 1788479, at *4 (D. Del. June 28, 2006); *Duhn Oil Tool, Inc. v. Cooper Cameron Corp.*, 474 F. Supp. 2d 1148, 1164 (E.D. Cal. 2007). The '985 and '906 claim limitations at issue are critically different from circumstances warranting invalidity for indefiniteness. Having reviewed all disputed claims, the Court concludes that the claims at issue do not impermissibly mix subject matter and retain a definite scope.

B.   *The Functional Language Argument*

Defendants next argue that if the claims are not means plus function claims, and they are not, then the claims are indefinite because they use purely functional language. Specifically, Defendants contend the language in the patents-in-suit "in order to cause," "software . . . operable to cause," and "for causing" is functional. Defendants argue that this functional language makes it impossible to discern what is actually being claimed. Plaintiff responds that the claims at issue are apparatus

claims that expressly designate functional limitations that are to be accomplished through a particular structure (namely a "computer readable program code for causing" or "software . . . operable to . . . cause").

As stated above, it is clear from the claims that what is claimed is an apparatus. That additional language exists in the claims describing the capabilities of each apparatus does not render the claims indefinite.

CONCLUSION

For the foregoing reasons, the Court interprets the claim language in this case in the manner set forth above. For ease of reference, the Court's claim interpretations are set forth in a table as Appendix A. Further, because Defendants have not presented clear and convincing evidence that: (1) claims 6–8 and 13 of the '906 patent; or (2) claim 28 of the '985 patent are invalid, the Court **DENIES** the motion for summary judgment.

So ORDERED and SIGNED this 22nd day of August, 2011.

_____
LEONARD DAVIS
UNITED STATES DISTRICT JUDGE

APPENDIX A

Claim Term	Court's Construction
executable application	a compiled program that is in native machine code in a format that can be loaded into memory and run
automatically [invoking/invoke] [the/said] executable application and executable application is automatically invoked by the browser	the executable application is launched without user activation
text format	coded information that describes how the content of a hypermedia document is to be displayed by a browser application
embed text format	coded information that specifies to a browser application that an object is to be embedded in a displayed hypermedia document
embed text format specifies the location of at least a portion of [an/said] object	No construction
embed text format, located at/corresponding to a first location in said first distributed hypermedia document	No construction
identifying an embed text format/an embed text format is identified	No construction
object	text, images, sound files, video data, documents, and/or other types of information that is presentable to a user of a computer system
[First] hypermedia document and distributed hypermedia document	a document that allows a user to click on images, sound icons, video icons, etc. that link to other objects of various media types, such as additional graphics, sound video, text or hypermedia or hypertext documents
file containing information	No construction

distributed application	an application that is capable of being broken up and performed among two or more computers
client workstation	as a computer system connected to a network that serves the role of an information requester
network server	a computer system that serves the role of an information provider